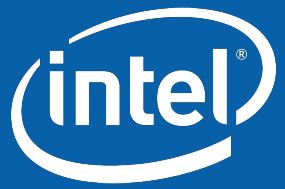


The Itanium® MCA Architecture

Cameron McNairy
Itanium® Processor Architect
Intel® Corporation

Agenda

Introduce the concept of system error detection, containment, and reporting

Defining processor, chipset, firmware, and operating system responsibilities

The Itanium® Machine Check Architecture

- Providing reliability, availability, and serviceability

Background and Motivation

Errors and failures happen and will become more likely

- Shrinking transistors lead to greater potential impact to an upset event
- Shrinking transistor operating voltage leads to greater potential that an upset will be sufficient to alter a transistor's value

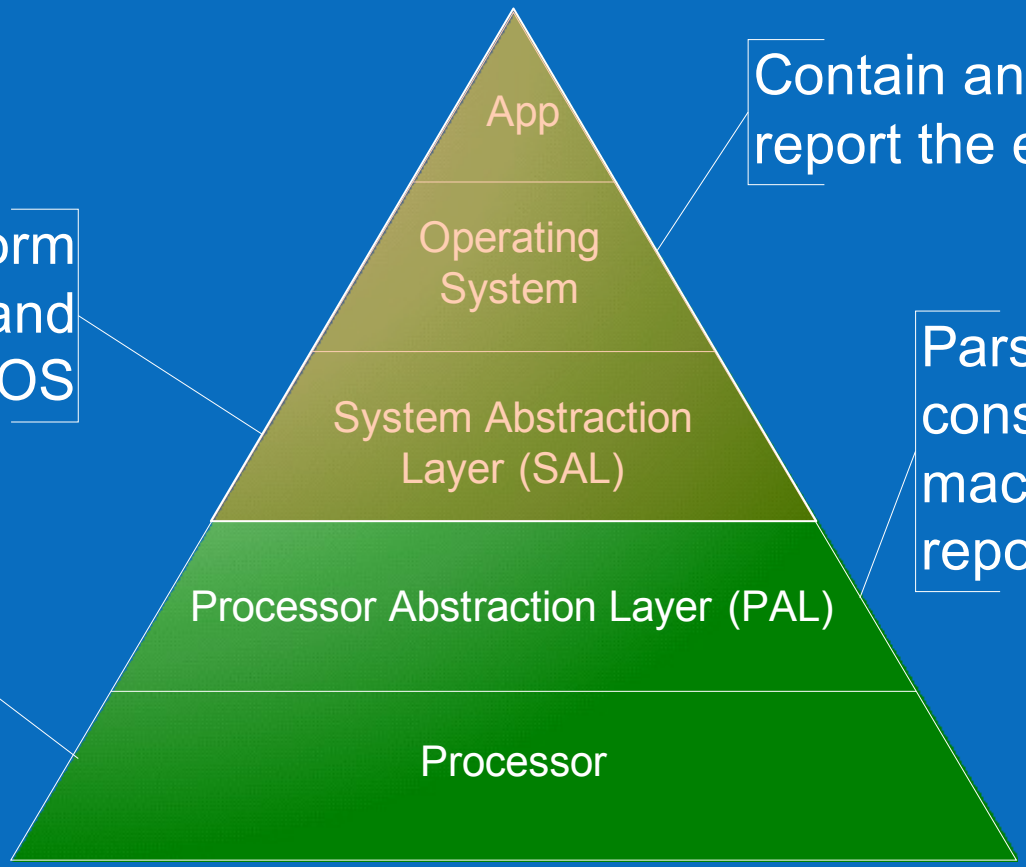
Error types

- Silent Data Corruption – an error occurred and escaped to memory or disk
- Detected and Uncorrected Error – an error occurred but was contained
- Corrected – an error occurred and was fixed

Reporting

- What happened to which component

RAS Partnership



Contain and report the error

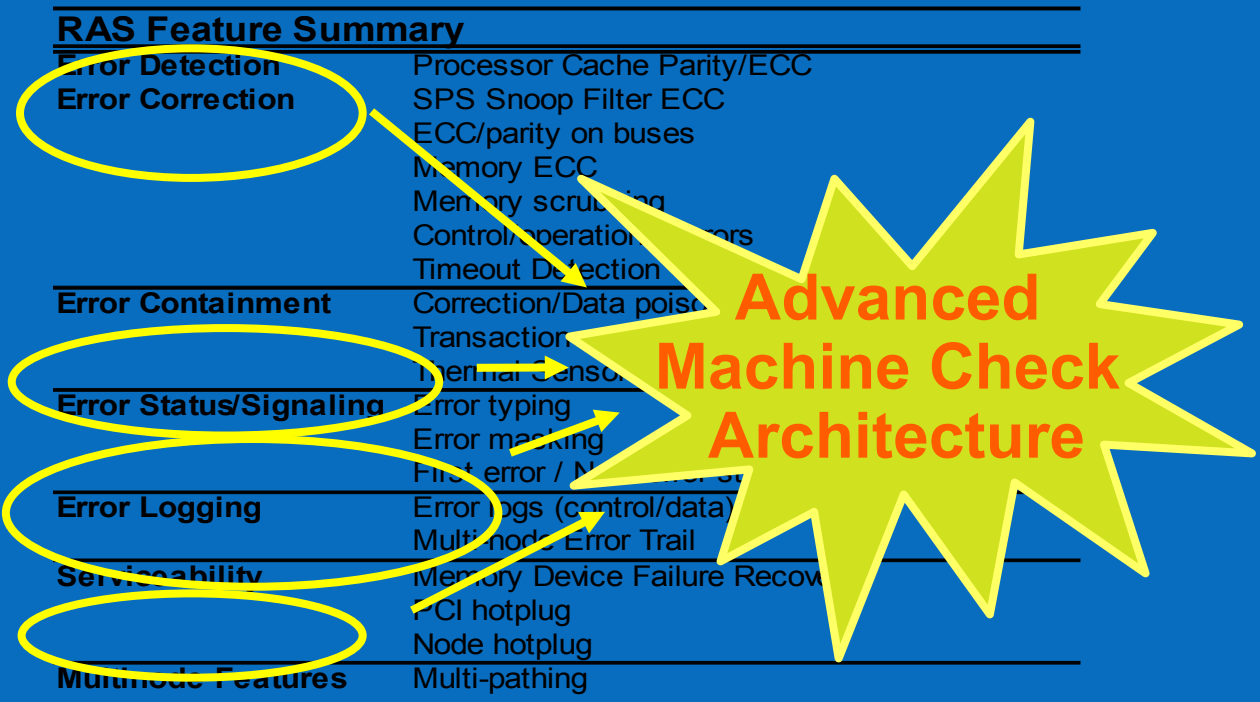
Collect platform errors and notify the OS

Parse error logs, construct consistent machine state, and report error severity

Detect, correct, log, and signal the error

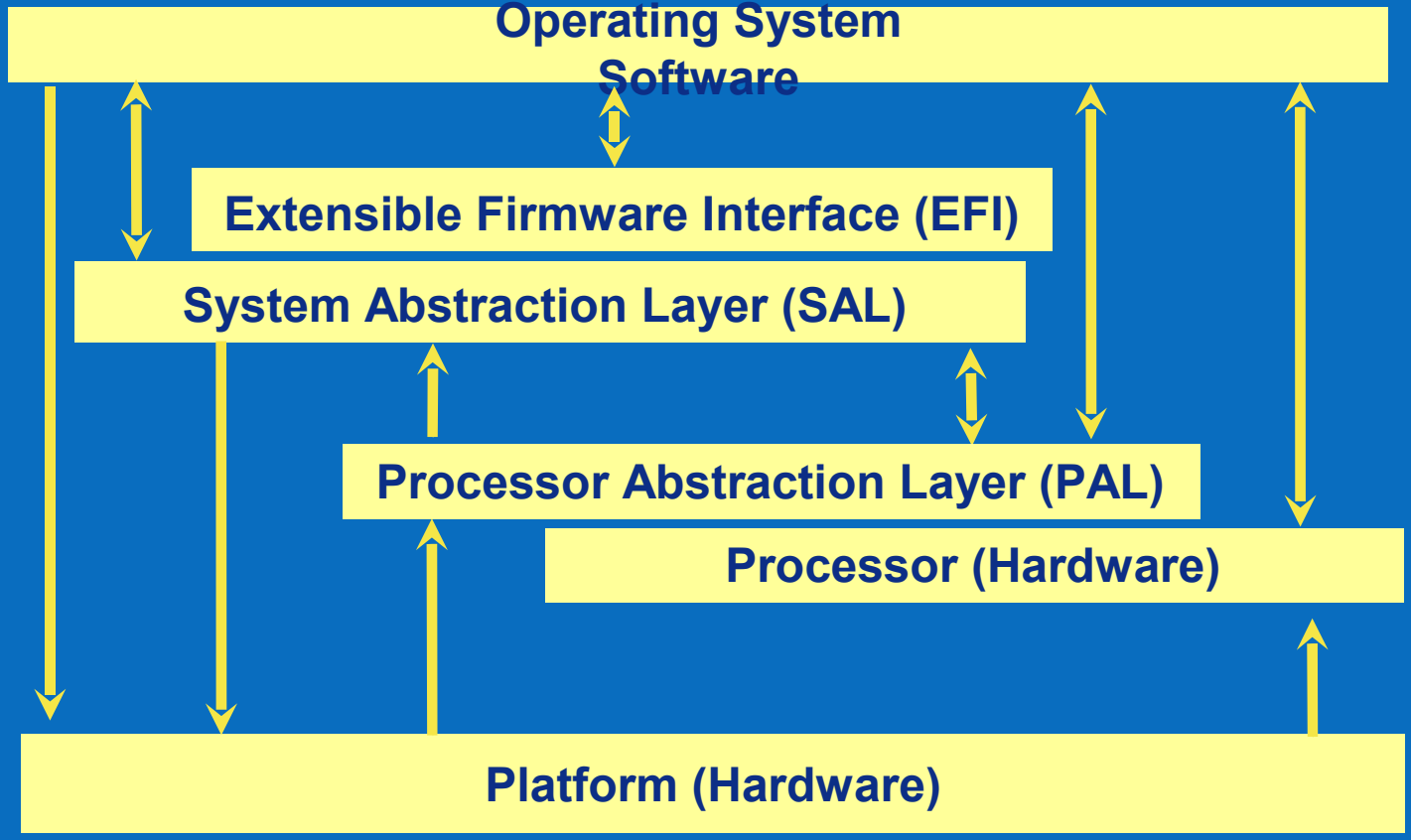
Itanium® Machine Check Architecture

Defines processor, chipset, firmware, and operating system responsibilities for advanced error handling



Itanium® Architecture Firmware

Architected APIs and error handling flows for hardware, PAL, SAL, and OS enables reliability innovation



Error Correction & Recovery

Hardware corrects most errors

Multilevel error handling for extended availability

Error Handling	Category
System reset 2-bit error in kernel	**Non-recoverable**
OS recoverable: System available 2-bit error in application	**Recoverable**
OS corrected: Execution continues Translation register error	
Firmware corrected: Execution continues Pellston cache reliability technology	**Corrected**
Hardware corrected: Execution continues Most 1-bit errors	

OS Error Recovery

Extends error handling capability from firmware to operating system

OS can correct errors if redundant information is available

- e.g., Translation registers

OS can terminate applications to contain errors and maintain system availability

- e.g., 2-bit memory errors

Flow defined by Itanium advanced machine check architecture

- Processor, platform, PAL, SAL, and OS responsibilities defined

Error Signaling

CMCI (Corrected Machine Check Interrupt)

- Low-priority interrupt
- OS may choose to ignore or poll, so processor *must* either scrub or signal MCA

MCA (Machine Check Abort)

- Synchronous or asynchronous
- Synchronous events are generally associated with core pipeline operations
- Processor must signal an error before data is consumed

BERR (Bus ERRor)

- A legacy error -- no Montecito processor errors signal this by default
- Causes all processors on the system interface to also take an MCA

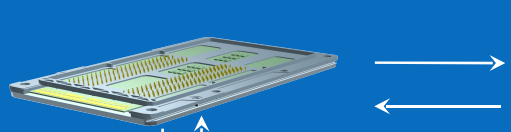
BINIT (Bus INITialization)

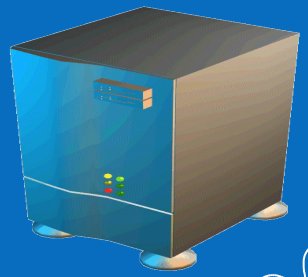
- Used to ensure containment
- Performs a very lightweight reset to clear queues, etc.
- Also used to break deadlock and guarantee forward progress
- Causes all processors on the system interface to also take a fatal MCA

IERR (Internal Error Detected)

- Used to ensure indicate that the processor has detected an error.
- Lockstep checker can use this to determine fatality of a divergence

Error Handling Model





Corrected

PAL_CHECK

SAL_CHECK

Error Logged

Did FW Fix Error?

No

Physical Addressing Mode Handoff

OS_MCA

Did OS fix Error?

Yes

OS Notified (interrupt)

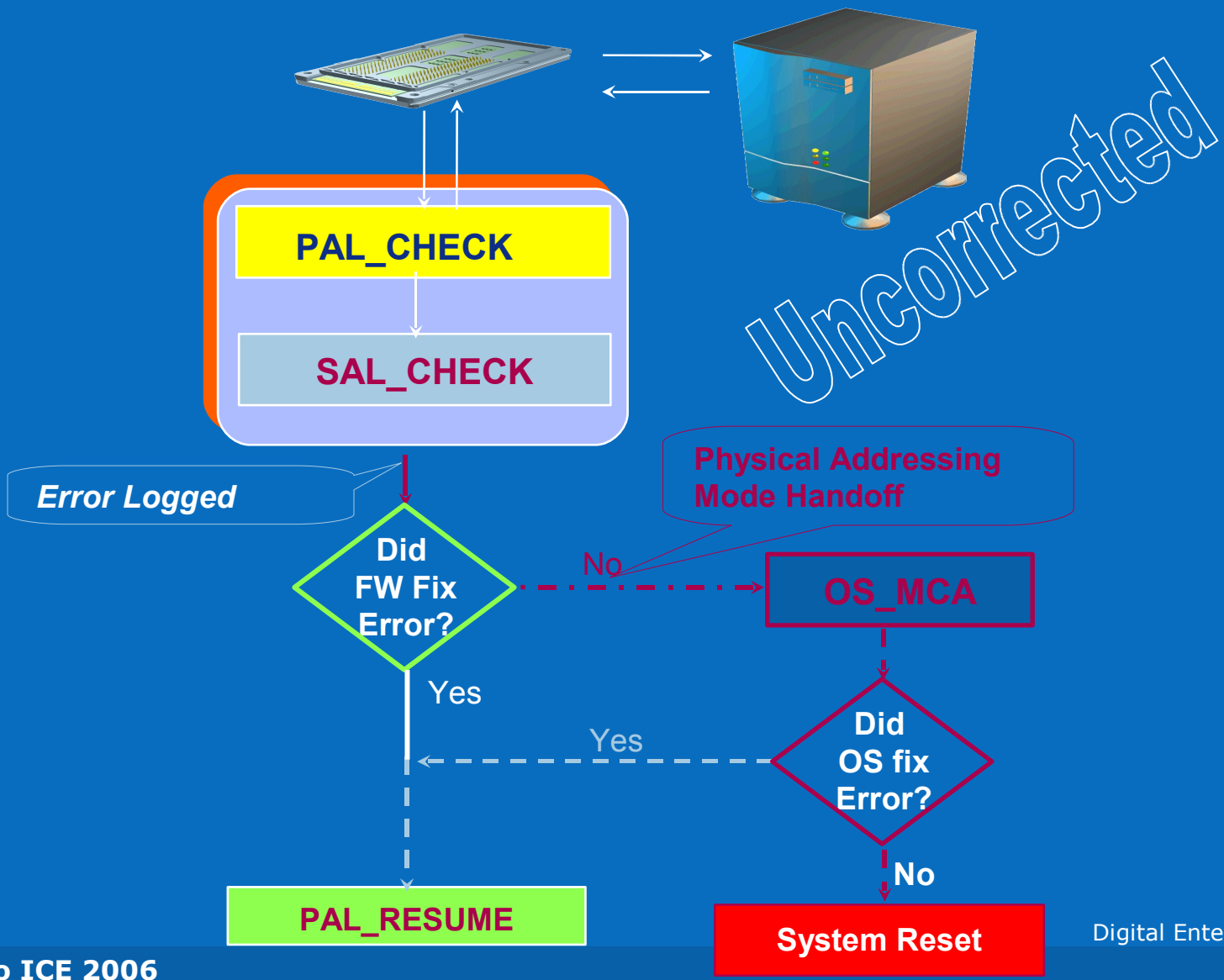
Yes

PAL_RESUME

No

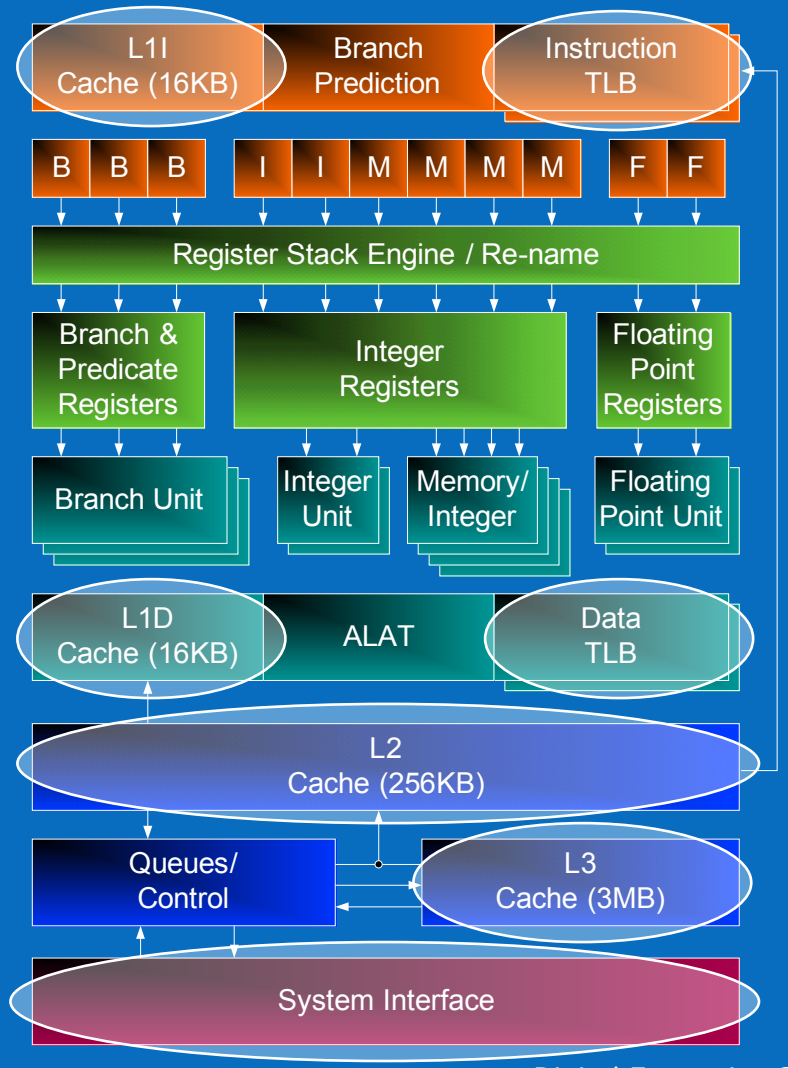
System Reset

Error Handling Model



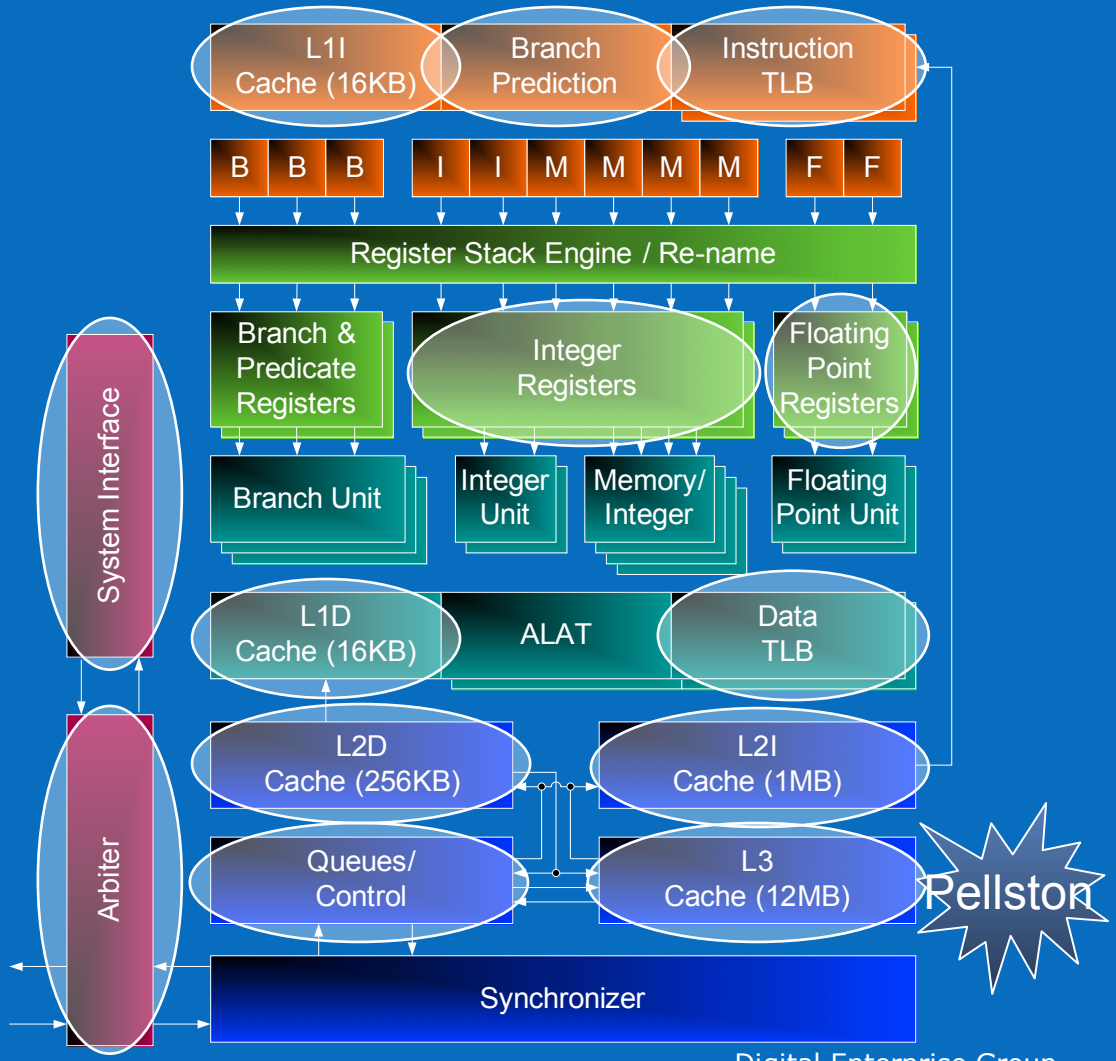
Itanium® 2 Processor Error Protection

- Correction (processor or PAL) for most arrays
- Detection for other exposed structures
- Mix of parity, error correction codes, multi-hit, and single bit transition approaches adapted to structure



Montecito Error Protection

- Builds on Itanium 2 protection
- Extends protection and correction capabilities
- Adds protection to performance structures to aid lockstep



Montecito Processor Error Coverage

Structure	Hardware	Action
L1 data cache	**Parity**	**PAL-correctable**
L1 tags	**Parity**	**PAL-correctable**
L2 cache data	**ECC**	**HW-correctable**
L2 cache tags	**ECC**	**HW-correctable**
L3 cache data	**ECC**	**HW-correctable + Pellston cache reliability**
L3 cache tags	**ECC**	**HW-correctable**
Register	**Parity**	**Recoverable**
TLB	**Parity**	**Recoverable**
Bus	**ECC**	**1-bit errors HW-correctable, 2-bit errors recoverable**

PAL-Correctable and Recoverable errors are dependent upon micro architectural state

PAL Error Handling Benefits

Provides advanced error handling

Pellston cache reliability technology

- PAL scrubs cache line to identify type of errors
- For soft errors, PAL invalidates line and resumes execution
- For hard errors, PAL disables line and resumes execution
- Enables improved reliability by removing cache failures from the system

Provides a programmatic interface for processor error information

- Provides abstraction from processor logging registers and improved handling of multiple errors
- PAL_MC_ERROR_INFO call provides detailed information on corrected, recoverable, and fatal errors

MCA Handling Overview

The PAL MCA handler is entered whenever the processor hardware detects an error event that it cannot automatically correct

MCA events are masked by the psr.mc bit only (not effected by the psr.i or psr.ic bit)

The PAL MCA entry address is fixed inside of PAL

MCA events pass control to the operating system if not corrected by firmware

OS can obtain further information about the error event by making firmware procedure calls

PAL Error Handling

PAL firmware support is a required component of the comprehensive error handling solution

The PAL

- Reports
- Categorizes
- Corrects subset of errors
- Offloads processor logic to reduce hardware complexity

PAL as an Interface

PAL provides a communication channel between the processor and software

Architected API allows same version of software to run on various processor implementations

- Abstracts micro-architectural details from higher level software

PAL fully recovers from selected error scenarios without required software interaction

- Optionally, PAL reports error information to higher level software

Intel Commitment to Reliability

Thousands of Intel engineers working on reliability analysis, design, and verification

- Transient errors, wear-out phenomena, and post-silicon reliability analysis

Itanium 2 processor error handling validation

- Over 30 validation engineer years of effort
- Automated validation of all hardware and firmware corrected errors, 2-bit memory and translation register recoverable errors, and Pellston technology
- Manual validation of recoverable, fatal errors, and lockstep
- Externally available error injection tool for the Itanium 2 processor and E8870 chipset

The most comprehensive error handling validation of any Intel processor